УДК 681.3.053:681.32:007.5

АНАЛИЗ ВЫПОЛНЕНИЯ ПОЛИТИКИ БЕЗОПАСНОСТИ В КОМПЬЮТЕРНЫХ СЕТЯХ: ПРОАКТИВНЫЙ ПОДХОД*

В.С. Богданов[1], И.В. Котенко[2]

В работе рассматриваются различные аспекты реализации перспективной интеллектуальной системы проактивного мониторинга (СПМ) выполнения политики безопасности в компьютерных сетях. Предлагаемый подход к реализации проактивного мониторинга основан на использовании всех доступных знаний о сети и текущей политике безопасности для автоматической генерации множества действий пользователя, которые способны опровергнуть или подтвердить выполнение заданной политики. Таким образом, имитируя действия пользователя, СПМ позволяет выявлять отклонения поведения компьютерной сети от политики безопасности на этапе эксплуатации. Предложена архитектура СПМ и обобщенная методика ее функционирования. Рассмотрены проблемы, связанные с проактивным мониторингом и методы их решения. Предложенный подход реализован в виде программного прототипа.

Введение

Основой для организации процесса защиты информации современных компьютерных систем является политика безопасности. Политика безопасности компьютерной сети, в частном случае, может быть описана в виде набора правил, каждое из которых разрешает или запрещает ту или иную операцию субъекта над объектом. При реализации компьютерной сети политика безопасности воплощается в конфигурациях оборудования и программных модулей. Эти конфигурации могут меняться в ходе функционирования сети, поэтому после ввода компьютерной сети в

* Работа выполнена при финансовой поддержке РФФИ (проект № 04-01-00167), программы фундаментальных исследований ОИТВС РАН (контракт № 3.2/03) и при частичной финансовой поддержке, осуществляемой в рамках проекта Евросоюза POSITIF (контракт IST-2002-002314).

[1] *199178, Санкт-Петербург, 14-я линия, д.39, СПИИРАН, bogdanov@comsec.spb.ru*

[2] *199178, Санкт-Петербург, 14-я линия, д.39, СПИИРАН, ivkote@comsec.spb.ru*

эксплуатацию требуется периодически подтверждать выполнение политики безопасности в ней и своевременно выявлять возможные несоответствия.

Предлагаемый подход и реализующая его система проактивного мониторинга (СПМ) основана на автоматической имитации действий пользователя, которые способны опровергнуть или подтвердить выполнение заданной политики безопасности [Богданов и др., 2005]. Генерация этих действий основывается на использовании знаний о выполняемой политике безопасности и конфигурации компьютерной сети, множестве возможных действий пользователя, условиях их реализации и последствиях выполнения, объектах и субъектах доступа, а также другой информации. В работах [El-Atawy et al., 2005] [Sailer et al., 2001] дано описание одного из способов реализации проактивного подхода применительно к тестированию брандмауэров и протокола IPSec. Настоящая работа посвящена применению этого подхода для тестирования политики безопасности в целом. Такое тестирование имеет ряд преимуществ по сравнению с формальным доказательством выполнимости политики безопасности. Приведем только два из них. Во-первых, оно позволяет убедиться в выполнении политики безопасности непосредственно на этапе эксплуатации компьютерной сети. Во-вторых, оно позволяет выяснить степень устойчивости политики к угрозам отказа в обслуживании, чего не позволяют адекватно реализовать подходы к тестированию, основанные на формальных моделях.

Однако применение проактивного подхода к тестированию компьютерных систем влечет за собой и ряд проблем. Во-первых, активная деятельность, направленная на проверку политики безопасности, сама может служить источником нарушения этой политики. В результате может быть нарушена конфиденциальность целостность или доступность информации, содержащейся в компьютерной системе. В работе рассматриваются и предлагаются пути решения этой проблемы: ограничение области тестирования, введение регламентного времени тестирования, принятие мер для сохранения целостности информации в ходе тестирования. Кроме того, проактивный подход к тестированию является очень трудоемким подходом, поскольку для того, чтобы полностью подтвердить выполнение политики безопасности, необходимо проверить результаты всех операций для всех пользователей над всеми объектами, что на практике практически невозможно. В работе [El-Atawy, 2005] рассмотрены три подхода к генерации тестовых примеров для тестирования правил политики брандмауэров: полный перебор, случайный перебор и выбор показательных данных. В нашей работе эти три подхода рассмотрены применительно к правилам политики безопасности общего вида.

1. Архитектура системы проактивного мониторинга

Обобщенная архитектура СПМ изображена на рис. 1. СПМ состоит из четырех основных компонентов: конфигуратора, сканера, сборщика информации и пользовательского интерфейса.

Конфигуратор — это компонент, который предназначен для планирования и формирования комплекса сценариев для проведения мониторинга политик. Он получает на вход спецификацию тестируемой системы и спецификацию проверяемых политик безопасности. На основе этих данных конфигуратор формирует сценарии для сканеров, находящихся в системе. Сценарии достаточно просты для выполнения сканерами без проведения дополнительного планирования. Данный подход имеет следующие достоинства: становится возможным планировать проверки, проводимые совместно несколькими сканерами; самая трудоемкая часть работы выполняется одним компонентом централизованно, сканеры представляют собой «легкие» компоненты с простой логикой исполнения; уменьшается поток информации, передающейся по сети, так как основная работа с базой данных действий пользователя совершается одним компонентом, а не несколькими.

Сканер — компонент, проверяющий определенную конфигуратором часть правил политики безопасности в заданном фрагменте тестируемой системы. Задание сканеру передается конфигуратором в виде сценария. Выполняя данный сценарий, сканер проводит проверку политики. Результаты проверки сканер отсылает сборщику информации.

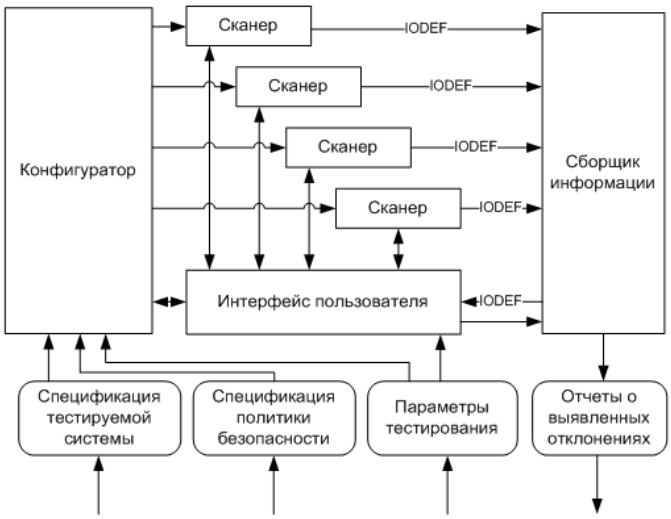

Рис. 1. Обобщенная архитектура СПМ

Сборщик информации получает отчеты о результатах проведенных проверок от сканеров, анализирует полученную информацию и формирует сводные отчеты, которые выдаются пользователю.

Интерфейс пользователя позволяет пользователю управлять работой всех компонентов системы, задавать входные данные конфигуратору, просматривать отчеты сборщика информации.

2. Методика функционирования СПМ

Входными данными для методики выступают спецификация тестируемой системы, спецификация проверяемых политик безопасности и параметры проверки, включающие подмножество правил политик для проверки, и параметры субъектов и объектов, для которых будет проводиться проверка. Для генерации сценариев проверки используется также информация о местоположении сканеров в системе. В качестве выходных данных служат отчеты о выявленных нарушениях выполнения политик. Методика задается в виде комплекса алгоритмов работы каждого компонента при проверке каждого класса политик. Охарактеризуем кратко сущность алгоритмов работы отдельных компонентов СПМ.

Общий алгоритм работы конфигуратора: (1) принять входные данные; (2) по спецификации тестируемой системы построить ее модель; (3) получить информацию о местоположении сканеров; (4) обозначить местоположение сканеров в тестируемой системе; (5) разделить правила политик на множества правил, для каждого из находящихся в системе сканеров по сегменту сети, для которого справедливы данные правила; (6) перебрать все множества правил, полученные на предыдущем шаге; (7) сформировать сценарий проверки данного множества правил для данного сканера: перебрать все правила из данного множества; добавить к сценарию проверку данного правила специфичным для политики алгоритмом; (8) передать сценарии сканерам для выполнения.

Общий алгоритм выполнения сценария проверки сканером: (1) выполнить проверки правил в порядке, предусмотренном сценарием; (2) в случае нахождения отклонений от правила добавить результаты проверки и условия, при которых возникают отклонения, к отчету сканера; (3) передать результаты проверки сборщику информации.

Общий алгоритм работы сборщика информации: (1) получить результаты проверки от сканеров; (2) обработать и обобщить результаты проверки, сформировать отчет о выявленных нарушениях; (3) выдать сформированный отчет.

В силу различия в политиках, для каждого класса политик необходимо использовать свой алгоритм проверки. Рассмотрим обобщенный алгоритм проверки для политики авторизации.

Алгоритм проверки политики авторизации: (1) перебрать всех пользователей; (2) перебрать все операции над активами; (3) перебрать все активы; (4) проверить успешность данной операции для данного пользователя над данным активом.

Существует четыре возможных исхода проверок: (1) если существует правило, разрешающее операцию, и операция выполнилась успешно, значит, политика не нарушена; (2) если существует правило, разрешающее операцию, и операция выполнилась неуспешно, значит, политика нарушена; (3) если не существует правила, разрешающего операцию, и операция выполнилась неуспешно, значит, политика не нарушена; (4) если не существует правила, разрешающего операцию, и операция выполнилась успешно, значит, политика нарушена.

Наиболее серьезным нарушением политики можно считать нарушение типа 4, поскольку оно свидетельствует о возможности выполнить операцию, фактически запрещенную политикой. Другой тип нарушения — тип 2, который свидетельствует о невозможности выполнить операцию, предусмотренную политикой.

3. Трудоемкость проверки

Для полноценной проверки выполнения правила политики безопасности, необходима полная проверка успешности/неуспешности всех допустимых операций над всеми активами, с использованием учетных записей всех пользователей системы. Такой подход очень трудоемок и не всегда позволяет проверить выполнение политики за приемлемое время. Существует ряд других подходов, которые позволяют с определенной долей вероятности утверждать, что политика выполнена, и лишены недостатков полного перебора. Рассмотрим два подхода, которые были реализованы в прототипе СПМ.

Первый подход («экспресс-анализ») состоит в извлечении необходимой информации из правила политики и проведении проверки для одного из представителей класса объектов проверки, выбранного случайным образом или по определенному правилу. Например, если правило политики формулируется следующим образом: «Администраторам компьютера разрешено создавать, модифицировать и удалять любые файлы в локальной файловой системе», то для проверки этого правила необходимо проверить возможность создания, модификации и удаления, любого, случайно выбранного, файла в системе, для случайно выбранного пользователя с ролью администратора. Такой подход подразумевает существенное сужение списка проверяемых операций там, где это возможно за счет анализа правил политики. Там же, где правило подразумевает любой объект какого-либо класса, проверка проводится для одного, случайно выбранного, объекта. Второй подход

является модификацией первого. При проверке выполнения политики для любого объекта какого-либо класса, второй подход выбирает не одного представителя данного класса, а несколько таких представителей.

4. Опасные операции

В процессе выполнения проверки политики на реальной сети может возникнуть необходимость выполнения «опасных операций». Под опасными операциями мы будем понимать операции, которые могут привести к нарушению целостности компьютерной системы и содержащейся в ней информации. Например, для файловой системы такими операциями будут удаление или изменение файла. Предотвратить нарушение целостности в результате выполнения таких операций поможет принятие следующих мер: (1) сохранение резервной копии информации, которая является объектом воздействия опасной операции; (2) проверка результатов опасной операции (с целью определения того, нарушена ли целостность информации); (3) восстановление поврежденной информации из резервной копии. Выполнение данных трех действий должно быть предусмотрено для любой из опасных операций.

Другим способом обхода опасных операций является введение нескольких режимов работы системы. Один их этих режимов позволяет проводить тестирование в полном объеме, включая опасные операции. Другой режим позволяет проводить тестирование всех операций, за исключением опасных. Опасные операции являются наиболее трудоемкими, так как требуют выполнения дополнительных действий. Поэтому использование безопасного режима также существенно увеличивает скорость работы системы.

Еще одним аспектом выполнения проверок является возможность возникновения конфликтов с работой других пользователей. Такие конфликты могут возникать в результате практически любых операций, как опасных, так и безопасных с точки зрения целостности информации. Например, сканеру может быть запрещен доступ на чтение к файлу, который заблокирован для редактирования пользователем. Одним из решений проблемы конфликтов может быть выделение специального времени для проверки системы, в которое пользователи не работают в системе, либо работают очень редко. Недостаток такого решения — невозможность полноценной проверки правил политик, которые имеют ограничения действия по времени. Другое решение — попытка определить причину конфликта и учитывать эту причину при проверке. Например, можно вести учет действий пользователя в системе, и при возникновении конфликта, проверять, не занят ли ресурс пользователем. Если ресурс занят, отложить его проверку или считать проверку пройденной, поскольку пользователь успешно получил доступ к ресурсу.

5. Прототип системы

Прототип СПМ был реализован с использованием Java. Основной экран интерфейса пользователя показан на рис. 2. Панель закладок на основном экране интерфейса пользователя позволяет просматривать информацию, касающуюся текущей загруженной конфигурации сети, текущей загруженной политики безопасности, состояния сканеров, а также отчет о выявленных отклонениях от политики безопасности. На рис. 3 показаны результаты работы системы в тестовой сети. Вкладка Report отображает отчеты о выявленных отклонениях от политики безопасности. С помощью вкладки Log можно просмотреть полный лог действий системы.

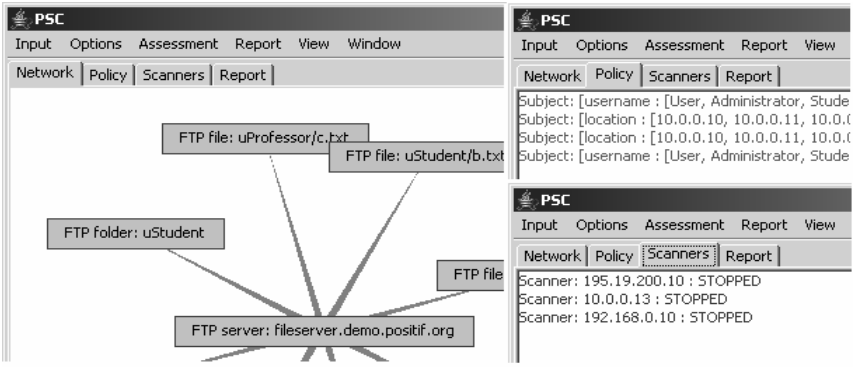

Рис. 2. Главный экран интерфейса пользователя

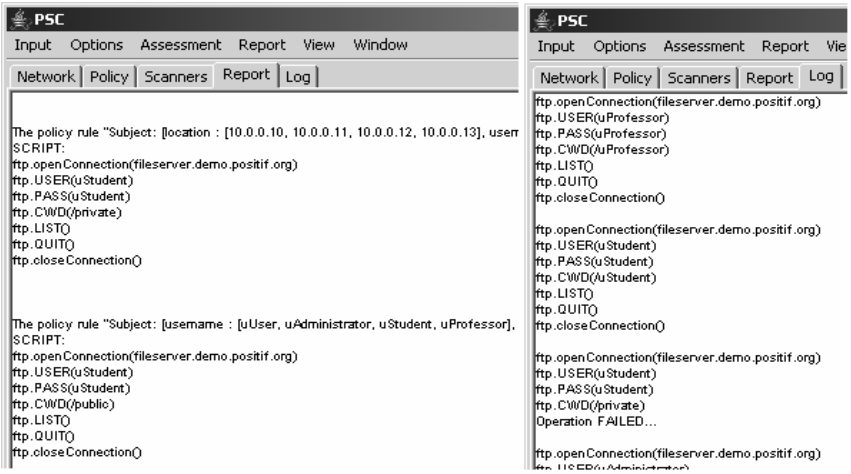

Рис. 3. Результаты работы системы

Заключение

В работе изложен подход и реализующая его система, предназначенные для автоматического проактивного мониторинга выполнения политики безопасности в компьютерных сетях. Предлагаемая методика проактивного мониторинга позволяет проводить проверки с различным уровнем точности и скорости: экспресс проверки, проверки со средней точностью и исчерпывающие проверки политики безопасности. Разработанные методики проверки отдельных политик учитывают уровень серьезности выявленных нарушений политики безопасности и позволяют производить проверку политики безопасности без выполнения “опасных” действий, то есть действий, которые могут нарушить целостность, доступность и конфиденциальность информационных ресурсов. В работе представлены меры предотвращения потерь информации в результате выполнения “опасных” действий в обычном режиме проверки. Рассмотрены возможные методы решения проблемы конфликтов взаимодействия между реальными пользователями и системой проактивного мониторинга.

Реализованный к настоящему времени прототип СПМ состоит из модулей четырех типов: (1) пользовательский интерфейс, позволяющий управлять поведением системы и просматривать результаты ее работы; (2) конфигуратор, координирующий действия сканирующих модулей; (3) сканеры, выполняющие действия, заданные конфигуратором из различных точек тестируемой сети; (4) сборщик информации, собирающий и обобщающий информацию, которая содержится в отчетах сканеров. Распределенная архитектура СПМ позволяет моделировать работу и взаимодействие нескольких пользователей.

Направлениями дальнейших исследований является совершенствование предложенных моделей и методик проактивного мониторинга и проведение экспериментальной оценки предложенных решений.

Список литературы

[Богданов и др., 2005] Богданов В.С., Котенко И.В. Архитектура, модели и методики функционирования системы проактивного мониторинга выполнения политики безопасности // Труды СПИИРАН, Выпуск 3, Том 1. СПб.: Наука, 2006.

[El-Atawy et al., 2005] El-Atawy A., Ibrahim K., Hamed H., Al-Shaer E. Policy Segmentation for Intelligent Firewall Testing // Proceedings of the First Workshop on Secure Network Protocols, Boston, Massachusetts, 2005.

[Sailer et al., 2001] Sailer R., Acharya A., Beigi M., Jennings R., Verma D. IPSECvalidate – A Tool to Validate IPSEC Configurations // Proceedings of the 15th Large Installation System Administration Conference, San Diego, CA, 2001.